NBdata

Marina Peñalver Ripoll

Contents

Introduction

In this document, we will use the `maSigPro` package to analyze time course data and identify significant genes.

First, we load the necessary libraries for the analysis.

We load the dataset provided by the `maSigPro` package, which includes the expression data (`NBdata`) and the experimental design (`NBdesign`).

```
data(NBdata)
data(NBdesign)
```

This dataset is part of a larger simulated and normalized dataset with 2 experimental groups, 6 timepoints and 3 replicates. Simulation has been done by using a negative binomial distribution. The first 20 genes are simulated with changes among time. Preview the first few rows of the data and the design matrix:

```
head(NBdata)
```

```
##       G1.T1.1 G1.T1.2 G1.T1.3 G1.T2.1 G1.T2.2 G1.T2.3 G1.T3.1 G1.T3.2 G1.T3.3
## Gene1      11       8      11       8       9       6      12       5      11
## Gene2       8      11       5       3      10       9       6      22       9
## Gene3      13       6       4       8      18       5      11      18       9
## Gene4       7      11      13       7       8      11       8      11      14
## Gene5      17       8       6       7       4       5      13      10      15
## Gene6      14      19       4      13       4      10       5      10      20
##       G1.T4.1 G1.T4.2 G1.T4.3 G1.T5.1 G1.T5.2 G1.T5.3 G1.T6.1 G1.T6.2 G1.T6.3
## Gene1       9       4      12      20      10       3       2       6      17
## Gene2       7       8       3       6       7       6       8       5       8
## Gene3       8      17       5       7       5       6       8       6       9
## Gene4       5       3      12       3      11       3       8       5       8
## Gene5       8      20       8       5       8      10       9      16      11
## Gene6      12      13      11       3       7       8       7      12      10
##       G2.T1.1 G2.T1.2 G2.T1.3 G2.T2.1 G2.T2.2 G2.T2.3 G2.T3.1 G2.T3.2 G2.T3.3
## Gene1      16       3      11      44      36      38      59      33      34
## Gene2       7       9       3      45      66      36      34      79      38
## Gene3       7       7      13      38      34      42      80      40      37
## Gene4      12      10       2      55      54      33      64      86      36
## Gene5       6       5      15      13      63      19      60      27      24
## Gene6       8      11       9      16      36      25      57      73      51
##       G2.T4.1 G2.T4.2 G2.T4.3 G2.T5.1 G2.T5.2 G2.T5.3 G2.T6.1 G2.T6.2 G2.T6.3
## Gene1      67      85      93      86      37     178     121     208      45
## Gene2     136      99      69     160      38      51     122     150     182
## Gene3      56      93      73     142     104     105     149     130     106
## Gene4      85      27     132      82     104      94      93     122     142
## Gene5      96      93      50      84      66      70      79     199     207
## Gene6      21     115      55      90     119      64      54      55     107
```

```r
head(NBdesign)
```

```
##         Time Replicates Group.1 Group.2
## G1.T1.1    0          1       1       0
## G1.T1.2    0          1       1       0
## G1.T1.3    0          1       1       0
## G1.T2.1   12          2       1       0
## G1.T2.2   12          2       1       0
## G1.T2.3   12          2       1       0
```

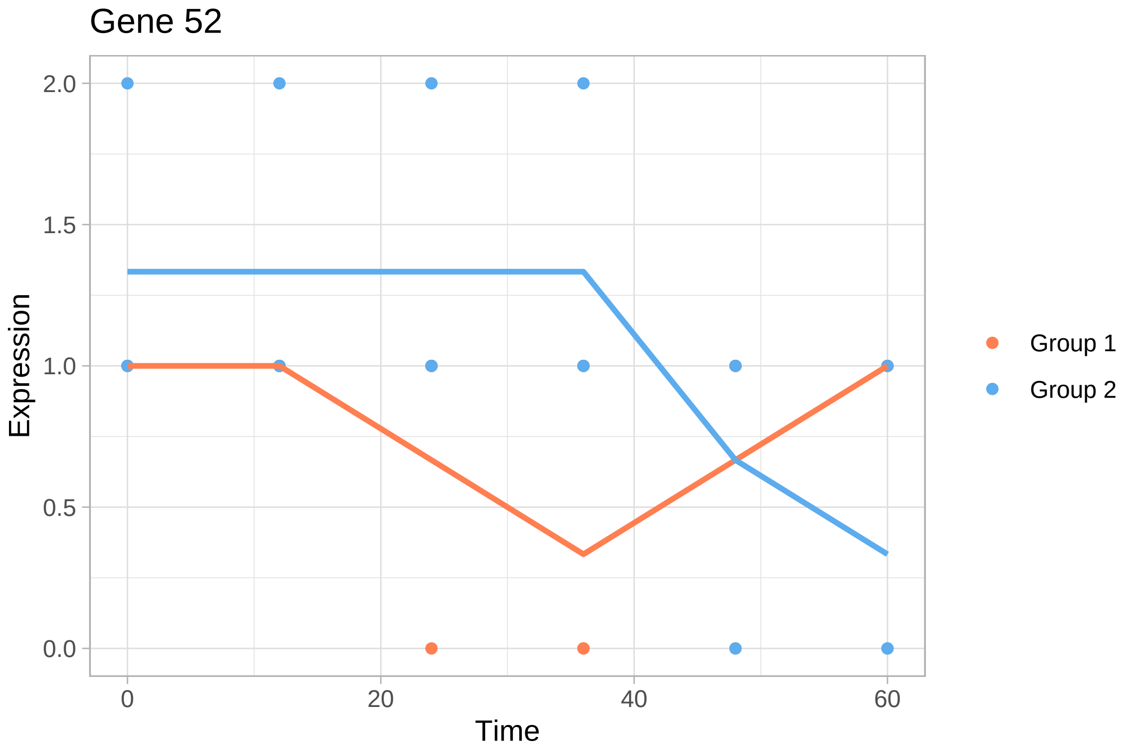
We create a design matrix from the experimental design.

```r
d <- make.design.matrix(NBdesign)
design <- as.data.frame(NBdesign)
```

Plotting Gene Expression

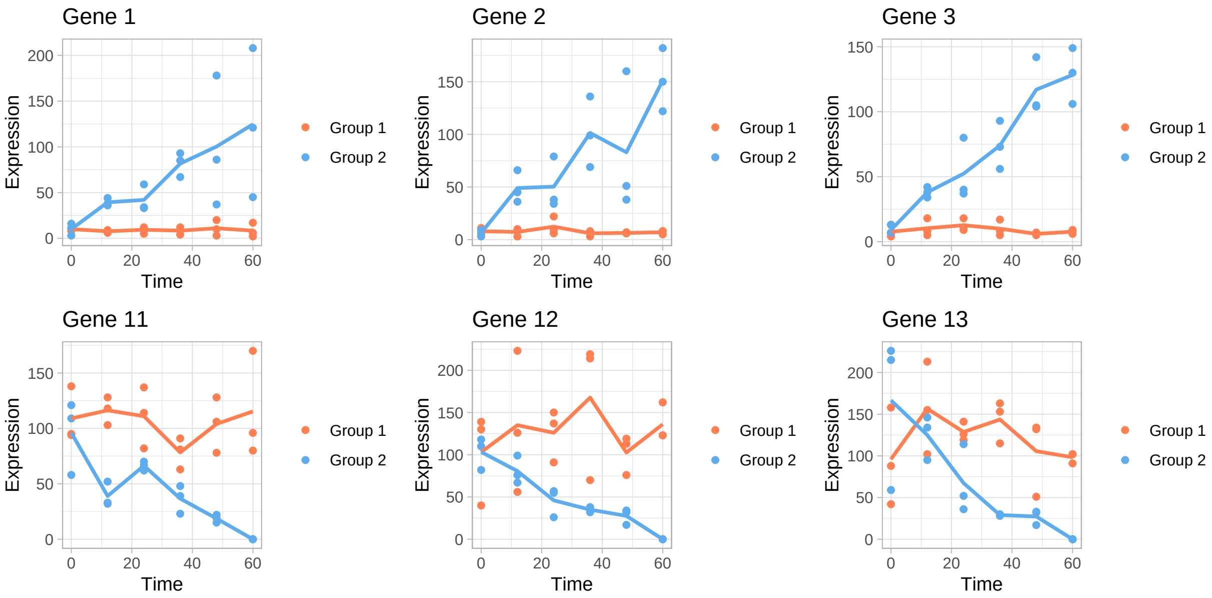
We define a function to plot the expression of a specific gene across time points for two groups.

```r
plot_gene <- function(gene_number, design, NBdata) {

  gene_data <- data.frame(
    Time = design$Time[1:18],
    Group1 = NBdata[gene_number, 1:18],
    Group2 = NBdata[gene_number, 19:36]
  )

  data_mean <- data.frame(
    Time = c(0, 12, 24, 36, 48, 60),
    Mean.G1 = sapply(split(gene_data$Group1, gene_data$Time), mean),
    Mean.G2 = sapply(split(gene_data$Group2, gene_data$Time), mean)
  )

  plot <- ggplot(gene_data) +
    geom_point(data=gene_data, aes(x = Time, y = Group1, color = "Group 1")) +
    geom_point(data=gene_data, aes(x = Time, y = Group2, color = "Group 2")) +
    geom_line(data=data_mean, aes(x = Time, y = Mean.G1), linewidth = 1,
              color="coral") +
    geom_line(data=data_mean, aes(x = Time, y = Mean.G2), linewidth = 1,
              color="steelblue2") +
    labs(title = paste("Gene", gene_number),
         x = "Time",
         y = "Expression",
         color = NULL) +
    scale_color_manual(values = c("Group 1" = "coral",
                                                       "Group 2" = "steelblue2")) +
    theme_light()
}

# Plot a specific gene
plot <- plot_gene(52, design, NBdata)
print(plot)
```



We define a function to combine plots of multiple genes into a single plot layout.

```
plot_genes <- function(gene_numbers, design, NBdata, ncol, nrow) {
  plots <- list()
  j = 1
  for (i in gene_numbers) {
    plots[[j]] <- plot_gene(i, design, NBdata)
    j = j + 1
  }
  combined_plot <- wrap_plots(plots, ncol=ncol, nrow=nrow)
  return(combined_plot)
}

# Combine plots of selected genes
combined_plot <- plot_genes(c(1:3, 11:13), design, NBdata, ncol = 3, nrow=2)
print(combined_plot)
```



Selecting Significant Genes

We perform regression fitting for each gene and identify significant genes.

- The `p.vector` function performs a regression fit for each gene, taking all variables present in the model given by a regression matrix, and returns a list of False Discovery Rate (FDR) corrected significant genes.

```
fit <- p.vector(NBdata, d, Q = 0.05, MT.adjust = "BH", min.obs = 20,
                family=poisson())
```

```
## [1] "fitting  gene 100 out of 100"
```

```
rownames(fit$SELEC)
```

```
##  [1] "Gene1"  "Gene2"  "Gene3"  "Gene4"  "Gene5"  "Gene6"  "Gene7"  "Gene8"
##  [9] "Gene9"  "Gene10" "Gene11" "Gene12" "Gene13" "Gene14" "Gene15" "Gene16"
## [17] "Gene17" "Gene18" "Gene19" "Gene20"
```

- The `T.fit` function performs stepwise regression to refine the model, using methods like backward elimination, and returns a list of significant gene profiles.

```
tstep <- T.fit(fit, step.method = "backward", alfa = 0.05, family=poisson())
```

```
## [1] "Influence: 12 genes with influential data at slot influ.info. Model validation for these ge
```

```
rownames(tstep$sig.profiles)
```

```
##  [1] "Gene1"  "Gene2"  "Gene3"  "Gene4"  "Gene5"  "Gene6"  "Gene7"  "Gene8"
##  [9] "Gene9"  "Gene10" "Gene11" "Gene12" "Gene13" "Gene14" "Gene15" "Gene16"
## [17] "Gene17" "Gene18" "Gene19" "Gene20"
```

- The `get.siggenes` function extracts the significant genes based on the R^2 value.

```
sigs <- get.siggenes(tstep, rsq = 0.7, vars = "groups")
sigs$summary
```

```
##    Group.1 Group.2vsGroup.1
## 1     Gene3              Gene1
## 2     Gene4              Gene2
## 3    Gene11              Gene3
## 4    Gene12              Gene4
## 5    Gene13              Gene5
## 6    Gene14              Gene6
## 7    Gene18              Gene7
## 8    Gene19              Gene8
## 9                        Gene9
## 10                      Gene10
## 11                      Gene11
## 12                      Gene12
## 13                      Gene13
## 14                      Gene14
## 15                      Gene18
## 16                      Gene19
```

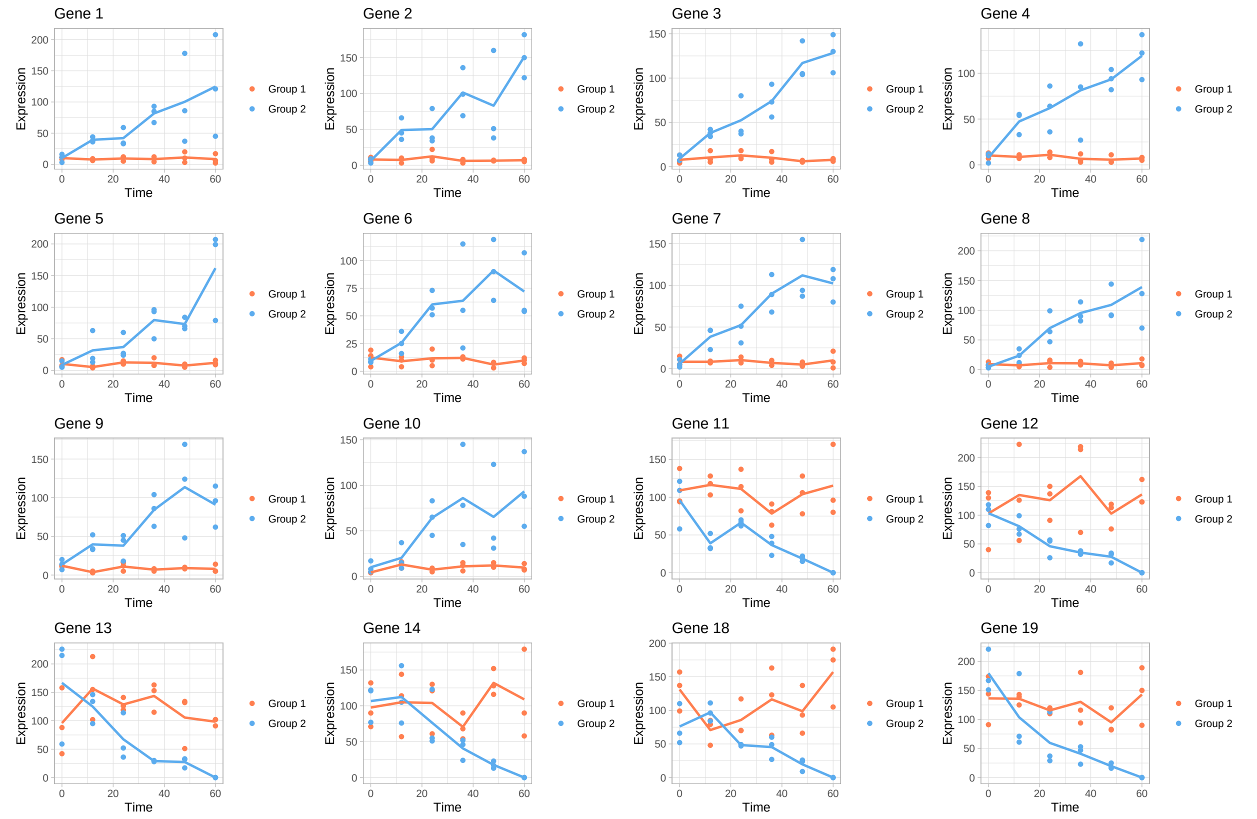
The value of the R^2 for each gene is:

```
tstep$sol$`R-squared`
```

```
##  [1] 0.8143910 0.8446812 0.9383557 0.8783471 0.8511779 0.8322692 0.9055105
##  [8] 0.8958995 0.8608001 0.7707060 0.7683814 0.7117493 0.7324734 0.7251357
## [15] 0.6432634 0.5579964 0.6431848 0.7713543 0.7853914 0.6864021
```

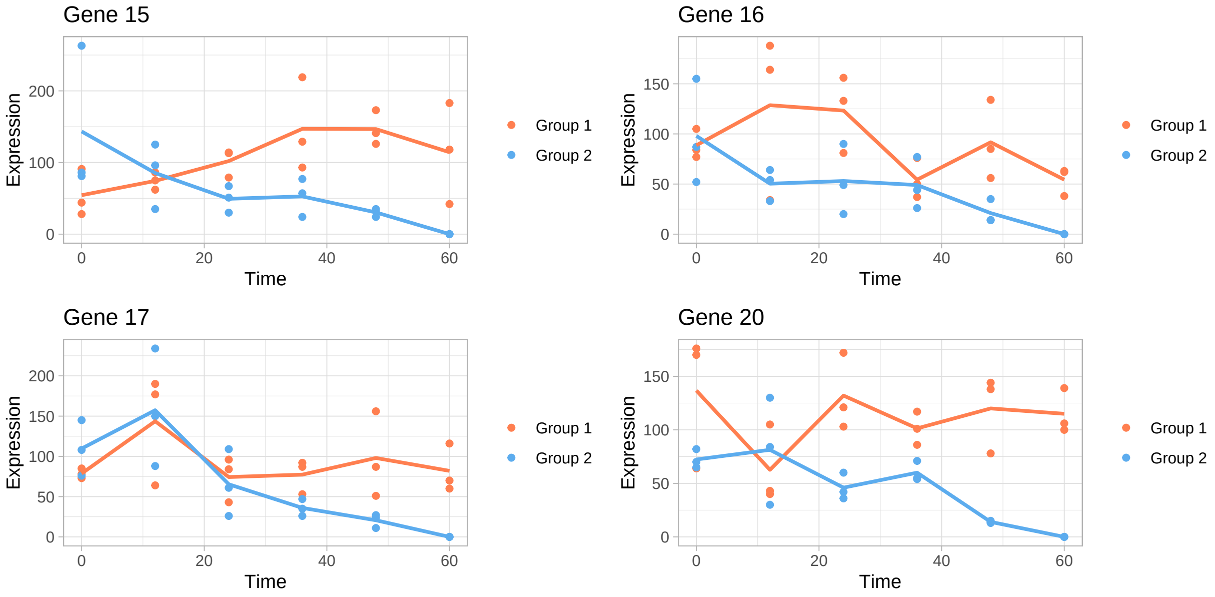
The significant genes are:

```
combined_plot <- plot_genes(c(1:14, 18, 19), design, NBdata, ncol = 4, nrow=4)
print(combined_plot)
```



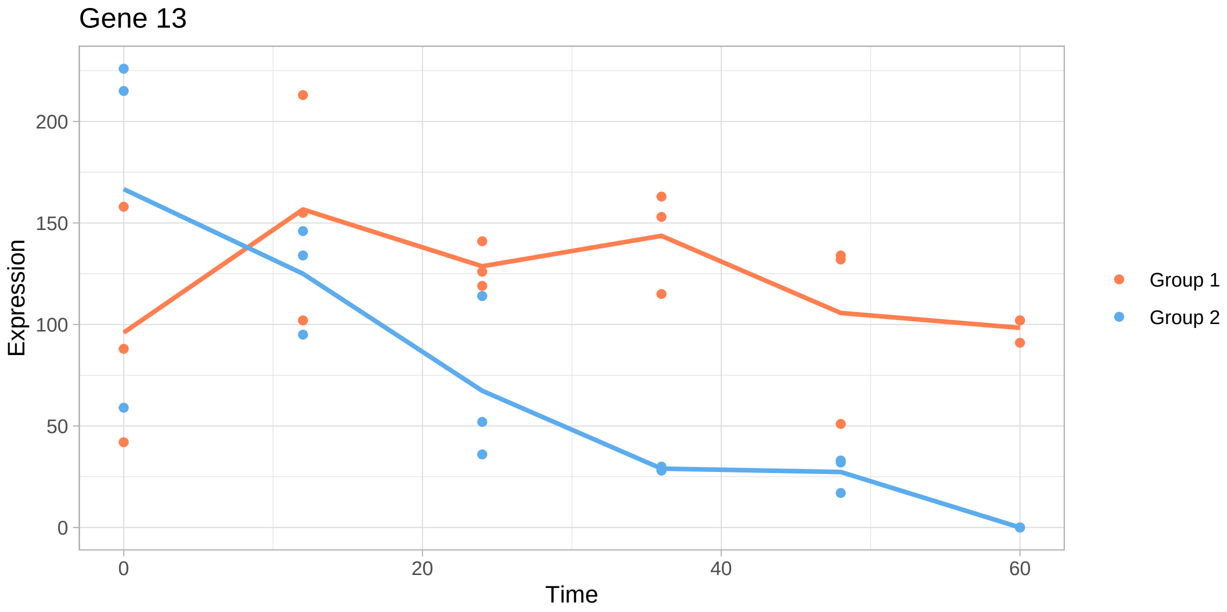
And the gene that we exclude are:

```
combined_plot <- plot_genes(c(15, 16, 17, 20), design, NBdata, ncol = 2, nrow=2)
print(combined_plot)
```



Finally, we select a gene for further study:

```
gen <- sample(c(1:14, 18, 19), 1)
plot <- plot_gene(gen, design, NBdata)
print(plot)
```


Gene 13
Expression
Time
● Group 1
● Group 2